A METHOD AND AN INSTALLATION FOR DOWNLOADING A USER DECODER PLATFORM

The present invention relates to the field of decoders used by subscribers to digital television, in particular when access is conditional.

Most digital television operators presently broadcasting in Europe offer decoders for hire. Such decoders enable all of the services of a given operator to be received. Abroad, decoders are already being sold by retail chains. However, each decoder is dedicated to a single operator or to a well-determined and unvarying group of operators. Consumers are not keen to buy such relatively expensive goods, particularly when they are not sure that they will enjoy the program pack offered by the operator or when they know that a purchased decoder will not be usable for receiving a pack that becomes available in the future.

The continuing increase in the number of television operators and in the additional services they provide, such as electronic program guides, pay-per-view, etc., make this situation less and less acceptable for the user.

The hardware platforms for decoders that receive satellite-broadcast television directly are standardized. ETSI's DVB standard requires all manufacturers to use a common hardware structure for decoders. In addition, it optionally provides for a common interface enabling modules for controlling access to different program packs to be connected in the form of PCMCIA cards suitable for insertion in a connector of a decoder. That solution is expensive. It requires numerous functions to be duplicated. Although it enables television broadcasts coming from a plurality of operators to be received by changing card, it generally does not give access to the associated services.

An additional difficulty lies in that the platforms for receiving a single pack can come from a variety of suppliers using different hardware, and only having a common application engine as imposed by the operator such as OPEN TV, MEDIA HIGHWAY, DAVID (digital audio-video interacting decoding), constituting a software layer at intermediate level. However, different operators generally require different application engines. Furthermore, a later version of the same platform can include additional features, giving access to services which at present remain inaccessible to people who possess earlier versions.

Documents US–A–5 440 632 and US–A–5 619 250, to which reference can be made, describe television terminals having respective platforms designed to download program updates for controlling the microprocessors in all terminals, in some terminals only, or in a single terminal. However those documents do not envisage the possibility of making it possible to switch from one operator to another.

The present invention seeks in particular to provide a method and apparatus making it possible for a decoder platform to be general-purpose, suitable for receiving broadcasts coming from different operators, regardless of whether they use the same access control mode and/or the same application engine.

To this end, the invention provides in particular a method of downloading application software specific to an operator into a general-purpose digital television decoder platform, in which:

· a secure boot loader is stored definitively in a protected and non-rewritable memory zone of the platform;

· a message identifying the platform and containing the application program making the platform suitable for decoding the data stream of the television signal of an operator and for processing its services is broadcast periodically in the digital television signal that comes from the operator and that is designed to be accessible, each of said messages having an electronic signature;

· on reception, the messages containing the program and identifying the platform and the operator are selected, decoded, and written into a rewritable program memory, optionally on user command.

Thus, either initially or during a subsequent download, the user can select an operator chosen from a list of operators who have made agreements with the decoder manufacturer, even though they might use languages (API) that are very different from one another for describing their applications or services.

Both in structure and in function, this method is completely different from merely downloading an update of supplementary software, reserved to subscribers of a single operator. It is also very different from merely transmitting messages for managing access authorizations, known as EMM. It makes it possible to access any one of a plurality of different packs using the same platform, and to do so in simple manner.

Two different situations can arise; both of them can be dealt with by implementing the invention.

The first situation is where the operator seeks to allow a user who already has a decoder to abandon the pack of a competitor in favor of the operator's own pack. Under these circumstances, the user cancels the subscription to the competitor's pack and subscribes to the new pack by a procedure that can be conventional, and requests downloading of the application software for the pack that is to be received. In the software as broadcast, the operator includes filter elements allowing only that particular owner of a platform to store the program. Thereafter, the user, e.g. by means of the remote control, calls the boot loader program which presents a menu enabling the user, again by means of the remote control, to input the parameters of the transponder for the pack that is to be received. The downloading process is then launched and its duration depends on the bandwidth allocated by the operator to this function in the broadcast.

The downloaded application software is written in a program memory. It can be a flash memory which takes a long time to write. For an operator who makes provision for this possibility only, it can suffice to transmit the application software giving access to the pack overnight only and in the form of successive packets transmitted at long intervals, which puts very little burden on the data rate available for television and other kinds of data.

The other situation is where competing operators wish to allow a common subscriber to jump between packs. Under such circumstances, an application program can be downloaded frequently in order to replace a program in memory. In order to avoid a wait that is too long (due to the time required to write flash memory), the programs are then stored and executed from program RAM that replaces or constitutes a front end for the flash memory. The presence of flash memory in addition to RAM makes it possible to conserve a version of the program in non-volatile form. In the absence of flash memory, downloading needs to be performed after each power interruption.

In addition to downloads performed at the initiative of the subscriber, it is possible to make provision for imposing updating downloads or function-adding downloads to take account of operating changes.

The method must satisfy two requirements. It must be selective, i.e. it must enable only certain platforms to be targeted; and it must be effective, making it possible within a given message to designate all of the platforms which are to receive the same version of the software.

These two functions can be performed by an operation that can be referred to as "filtering", which consists in specifying the decoders concerned by a given data stream by means of indications written either in the header of a software download stream or in the information tables associated with services (PSI and SI). For this purpose, the header (or the PSIs or the SIs) can include a plurality of fields defining characteristics which are also recorded in the platforms. These characteristics can be unchanging, such as those of the hardware portion, and others can be changing, such as those of the software portion.

The invention also proposes an installation for downloading application software into digital television decoder platforms, the installation comprising:

· in each platform, a general-purpose processor module that is independent of any operator and that serves: to select and extract a data stream representing application software specific to the program pack offered by an operator, to record it in a rewritable program memory for storing said software, and to control the decoder to implement the services identified by the software; and

· with the broadcaster, means for inserting in repetitive manner in the broadcast digital data stream both a sequence of blocks representing said specific software and information describing the characteristics of only those decoders that are to be loaded.

The selection and extraction means can be constituted by a general-purpose processor module that is independent of the operator for performing all of the functions.

In a variant embodiment, all or part of the program (or of software giving access to the program) can be transmitted over the telephone network, providing the platform includes means for being coupled thereto. Nevertheless, this complication is generally not necessary since the bandwidth required for transmitting an application in a reasonable length of time remains small. For example, if an operator is using a satellite channel with a bandwidth of 36 MHz together with four transponders, only 1% of the available data rate, i.e. about 1.2 Mbits/s needs to be given over to downloading a piece of software of average length, 1 MByte, in about 8 seconds.

If program changing is expected to be exceptional, for the purpose of changing subscription, then transmission can be performed at an average data rate that is extremely low and that will have no perceptible influence on the available bandwidth.

The above characteristics and others will appear better on reading the following description of a particular embodiment, given by way of non-limiting example. The description refers to the accompanying drawings, in which:

· Figure 1 is a block diagram showing the hardware architecture of a platform comprising a decoder associated with a television set;

· Figure 2 is a diagram showing downloading;

· Figure 3 shows one possible header structure (or private descriptor in a PSI or an SI table) for filtering purposes;

· Figure 4 shows a loading sequence; and

· Figure 5 is a diagram showing one possible way of managing keys.

The invention is described essentially in its application to a decoder for receiving digital television signals of the MPEG2 type, constituted by a multiplex made up of successive packets. The packets convey:

· the audio and video components; and

· digital data, including the software to be downloaded.

The architecture of a decoder platform is generally as shown diagrammatically in Figure 1. It comprises:

· a network interface 10 performing reception and demodulation functions, and of structure that depends on the network (cable network, satellite direct broadcast, terrestrial broadcast network);

· a time demultiplexer 12 which also performs unscrambling, for separating the components of the received signal;

· audio and video decoders 14 and 16; and

· a data processing and decoder management module 18.

The operation of the demultiplexer 12 depends on the module 18. It serves to direct video packets to the video decoder 14, audio packets to the audio decoder 16, and data to the module 18. It unscrambles the components that have been scrambled on transmission to control access.

The audio and video decoders 14 and 16 perform MPEG2 decompression and deliver the decompressed digital information to digital-to-analog converters 20 and 22 which output audio and video signals usable by a television set.

The module 18 manages all of the elements that are internal to the decoder and also user interface elements 24 such as a keypad, a remote control infrared receiver 23, and a display. It can also drive an input/output interface 25 connected to optional elements suitable for extending the facilities available, such as a telephone modem 26 or a high-speed interface 28 for connection to a microcomputer. The processor is also generally connected to a connector 29 for receiving a microcircuit card or smart card, e.g. containing circuits for computing an unscrambling key.

The module 18 has a processor 30 connected by a bus 32 to memories. In conventional manner, these memories comprise:

· a read-only memory or ROM 34 which is not volatile and not reprogrammable without hardware intervention, which memory is directly accessible by the processor;

· a volatile working memory or RAM 36, directly addressable by the processor and intended for manipulating data.

To enable the invention to be implemented, the memories also include additional memory spaces serving in particular to store:

· a loader program for initialization and starting purposes, referred to as a "boot loader", situated in a memory zone that is not volatile, protected, and not rewritable (the non-rewritable nature of this zone can be obtained, for example, by masking during manufacture); and

· the complete operating software for a digital program pack specific to a private operator, with this being in a zone that is rewritable.

In the example shown in Figure 1, the memories comprise, for this purpose:

· a reprogrammable non-volatile memory 38 that is directly accessible by the processor, serving to receive the application software, e.g. a flash memory; this memory can be designed to store the programs specific to a plurality of packs if the platform is designed to make it possible to jump between packs without having to wait for re-loading; in general, it contains the operating software;

· a non-volatile memory 40 designed to receive configuration data for the decoder; this memory which is not necessarily addressable by the processor can be an electrically reprogrammable read-only memory or EEPROM.

The ROM 34 can be a non-modifiable portion of the memory 38, if the memory 38 is a flash memory.

The software architecture of the decoder can be considered as having three functional levels or layers, the driver layer, the system layer, and the interactive application layer.

The driver layer is specific and matches the hardware architecture. It is this layer which makes it possible to perform the hardware functions provided by the decoder.

The system layer manages the platform and provides the general services, including the application engine, that are required to enable it to operate, and also the services that are called by the interactive applications. To perform this function, the system layer generally has an interpreter, serving to transform source code into object code. However a compiler is not necessary since it suffices if transformation is performed on each new use of the system layer.

Finally, the interactive application layer provides local interactivity and makes use of the application engine; it can also be designed to constitute the interface with the modem 26 for connection to a telephone line. This layer has user interface applications which call on services provided by the system layer.

The applications and the associated resources are partially resident, i.e. stored in permanent manner in the ROM of the decoder, and they are partially downloaded by the system layer from the MPEG2 standard television signal.

The user interface applications are generally written in a script language. The system layer interprets the script language information and manages activation and downloading of interactive applications. This system layer is loaded into the platform in the form of a code that can be interpreted directly by the processor 30.

Switching from one pack to another corresponds mainly to reconfiguring the memories.

Downloading operations

An application program is downloaded as follows.

Changing pack implies loading all of the software enabling the pack to be processed, and this is independent of any special features concerning access control mode.

For this purpose, it is necessary to load or change software residing in the decoder, which is done by reinitializing all of the program memory 38, which is generally a flash memory.

The data which is transmitted to the platform during downloading to reinitialize the flash memory 38 is the same for all platforms having the same hardware structure.

The diagram of Figure 2 corresponds to downloading making use of the data portion of the broadcast stream. The software to be loaded is in the form of a file. Within the platform, it is extracted and sent to RAM 36 where it is reassembled prior to being written in the program memory 38 which will thus end up with the driver layer, the system layer, and the application layer, including the application engine.

Under other circumstances, downloading can take place via the input/output interface 25, using a modem or a microcomputer.

Under all circumstances, downloading implies, at the broadcaster, generating image files for writing in the program memory 38 of the platform. These files can be of a very wide variety of kinds:

· already-compiled object files;

· applications written in script language;

· other functions such as a library function.

The "image" files as constituted in this way are then formatted to adapt them to the method of transmission that is to be used, i.e. either over the television program broadcast network or else over the wire network.

In both cases, the first operation performed in the platform, on receiving files, is selectivity filtering so that only those applications programs which come from a specific program supplier are loaded. As explained below, this operation can be accompanied by checking an electronic signature in the header of the data stream constituting the application software to be loaded.

Filtering

Selectivity filtering makes it possible to ensure that the application program is loaded into identified platforms only, and to ensure that it is loaded into all such platforms. At any given moment, there exists numerous types of platforms that are in operation, and as a general rule they contain different software. Even if they are of different types, platforms that are initially intended for a given operator or program supplier will all have the same application engine. However the application engine changes on switching from a platform programmed to receive the pack from a particular supplier or operator to a platform programmed for another operator: it therefore needs to be replaced in the application memory.

Depending on the origin of the decoder and the hardware architecture of the decoder, the elements which can change include the following:

· the manufacturer of the decoder, where manufacturers often make use of proprietary architecture;

· decoder acquisition mode (rental, purchase, purchase with a subsidy dedicating the decoder to a particular operator for a determined duration) which can give rise to different access control functions and thus to different system layers;

· date of acquisition, since the software might have been modified over time.

All of these elements are included in an identifier of the decoder, which identifier can include the following fields, in particular:

C_1: manufacturer identifier;

C_2: version of the hardware software;

C_3: acquisition mode (rental, subsidized sale, non-subsidized sale, etc.);

C_4: software identifier, specifying the version of the software currently loaded in the decoder;

C_5: individual serial number of the decoder.

Unlike the others, the field C_4 will be changed on each download.

To make filtering possible, an identifier is provided in each decoder, and each data stream representing application software includes parameters enabling reloading or updating operations to be performed only in appropriate decoders.

The header will include respective fields allocated to each of these parameters.

By way of example, Figure 3 shows one possible structure for the header of a data stream; this header is constituted by a block of N bytes, preceded by a block specifying the number N.

Each field of the decoder corresponds either to a single selection filter specified by the corresponding field of the header, or else to a plurality. Loading can take place in a decoder only when all of the filtering operations give rise to a positive result.

The first field C_1 can be limited to a single filter F_1 recorded in ROM, specifying the manufacturer concerned by means of an identity number ID.

The second field C_2 can comprise a plurality of filters, corresponding to different versions of the platform, and a filtering operator constituted by an OR function: for the filtering result to be positive, it suffices that one of the filters F_{2i} recorded in ROM should match C_2.

The field C_3 can be constituted by a single filter F_3, with the filtering operator then being an intersection. The result of filtering is positive if $C_3 \wedge F_3$ is non-zero.

The field C_4 has a single filter F_4, and the filtering operator is then the comparison operation $C_4 < F_4$:

loading needs to take place in all decoders that have not yet been updated.

The field C_5 is generally longer than the others, and comprises 32 bits, for example; e.g. it will contain a plurality of filters F_{5j} each giving a bottom limit and a top limit, identifying a series of decoders for which updating should be performed. The result of filtering is positive if the value contained in the field C_5 of the identifier lies between the two values given by at least one of the filters F_{5j}.

The field C_6 specifies the operator (or the operators) with whom a subscription has been taken out. It has one or more filters F_6 recorded in rewritable memory.

Addressing

The data to be written in the application memory 38 is transmitted to the decoder with an indication of the addresses at which it is to be copied into the memory 38.

It can happen, particularly when a formatting RAM 36 is located upstream from the program memory 38, that the data for a complete program cannot be acquired in a single operation or using a single address.

Under such circumstances, the data representative of the software to be downloaded is transmitted to the decoder in the form of successive blocks of contiguous data, and the data of any one block is copied into the same address in the program memory 38. The loading of software into the program memory 38 can then be sequenced in the manner shown diagrammatically in Figure 4. Each of the successive data blocks has a starting address A_1, ..., A_n specifying an address in the program memory 38, followed by a data portion D_1, ..., D_n, and an error correcting code. These blocks are preceded by transmission of a header block 44 having an application descriptor DA and descriptors DD_1, ..., DD_n for the successive blocks. The starting addresses make it possible for writing to take place immediately in the program memory 38.

The header block identifies the application to be loaded and lists the blocks that make it up. The data blocks making up the application are managed on the basis of image blocks which have transport security information added thereto constituted by a code for detecting (and optionally correcting) any errors. This can be constituted in particular by a cyclic redundancy check, generally referred to by the abbreviation CRC.

In practice, when a subscription is being loaded, downloading takes place as follows. After switching on, the user starts the downloader program by pressing one or more keys of the remote control. This program presents a menu enabling the user to input parameters of the home transponder, and of the new program pack (at least frequency, polarization, error correction code rates, and symbol rate) by means of the remote control. To make this task easier, this information can be input in compact form, e.g. in the form of a few decimal digits given by the operator when the subscription is taken out. Pressing on the confirm key then launches downloading. This downloading operation relies on the monitoring and selection functions that use the fields C_1-C_6. The following take place:

· the operator, the version number, and the manufacturer are checked;

· the version number, the manufacturer, the serial number are selected, with selection being possible without requiring an authentication process.

As mentioned above, downloading is made secure so as to prevent:

· downloading of data that is not transmitted by an authorized operator;

· downloading of data into a platform that is not authorized to receive it.

Security can be based on encryption using private and/or public keys. It is known that public key encryption uses an algorithm that is difficult to reverse, such that knowledge of the public key and of the encoded message does not suffice to return to the original message without performing calculations that will take an unrealistic length of time.

In Figure 4, dashed lines show additions to be provided to the header 44 so as to make the message secure.

Each data block is associated with a signature S_1, ..., S_n which is included in the header. The signature, as calculated from the data of the corresponding block, serves to verify that the block is authentic.

In addition, the header has a signature which is transmitted in encrypted form S. The encryption algorithm for the signature of the header block is a private key algorithm, e.g. of the RSA type. The private key is known only to the manufacturer. The non-encrypted signature is calculated on the basis of the encrypted signature S in the decoder by means of a public key algorithm stored in the ROM 34 or in a protected zone of the program memory 38, if it is a flash memory.

The signature S serves to verify the authenticity of the header block, and thus of the data that it carries, and in particular of the signatures S_1, ..., S_n.

The way in which keys and functions are shared when a plurality of operators 1, ..., i are grouped together to use common private keys, can be as shown in Figure 5. On the basis of common private keys, the operators give the manufacturer of the decoder software public keys which are written into the ROM 34 at the same time as the filters F_1, F_{2i}, F_{5j}.

The instructions for booting the decoder when it is put into operation are also stored in ROM, together with the updating loader of the terminal. To mitigate cases of corruption in the program memory 38, particularly if it is a flash memory, due to an interruption occurring during loading, the updating function of the terminal is directly associated with the boot function of the decoder processor in the event of corruption being observed.

The invention makes it possible to allow relationships between operators and users to change in simple manner. Because the operator identifier is stored in flash memory, unlike the other parameters which are stored in ROM, it is possible to reallocate a hired decoder when it is returned. A decoder can be "freed" of any connection with any particular operator. Selection is performed by logic operations that are simple and that can be implicit by default.

In the particular circumstance of broadcasting using the MPEG2 standard, the data for updating and loading application software is conveyed in a private data DVD service of the type specified in the standard as "terminal update". The blocks constituting the software to be loaded are split up into elements having a maximum size of 4064 bytes, each element having a 16-byte header. A service for updating or reloading software is identified on the basis of network signalling data.

The method of the invention for downloading application software does not interfere in any way with downloading software updates from the current operator, i.e. the operator with whom the user has taken out a subscription.